100th Anniversary of
Chula Engineering 2013

String & File Processing

python
powered

Topics

- Revision
- Escape Characters
- String Methods
- Function composition & Method Chaining
- Reading from a file
- Writing to a file

Revision

```
s = "I'm a string"
t = 'I said "This is a string".'
chars = input() #key input are strings
c = 0
for ch in chars: #for each ch in chars
    if ch in s:
        c += 1
for i in range(len(t)): #for each char in t
    if t[i] in s:
        c += 1
r = ""
for k in range(2,10,2):
    r += str(k)    # 2468 string concat
r = 2*r            # 24682468 repetition
```

Revision Exercise

See Colab Notebook

Revision Exercise: Write a program to transform any word (lower case) to its plural form.

The rule (this is just a simplified rule) is :

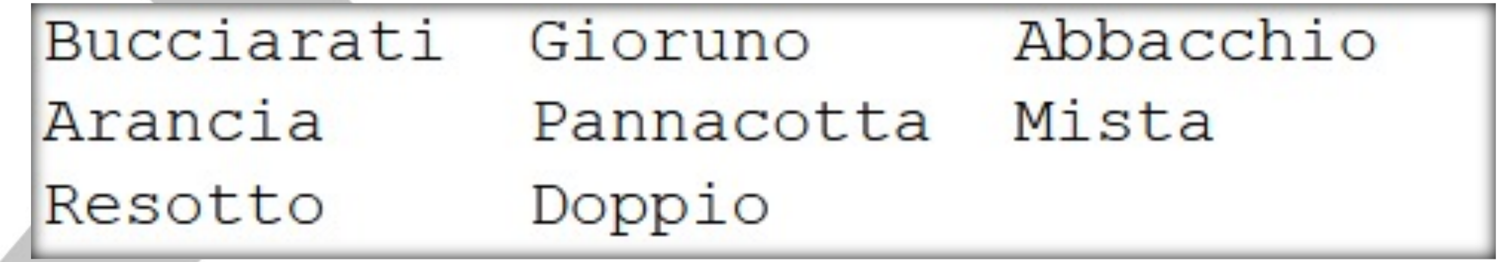
- If the word ends with "s", "x", or "ch" then add "es".
- If the word ends with "y", but the character before "y" is not a vowel, then change "y" to "ies".
- Otherwise, add "s" to the back of the word.

Example 1

Read N names (each name <= 10 characters), separated by ", " and display 3 names on each line.

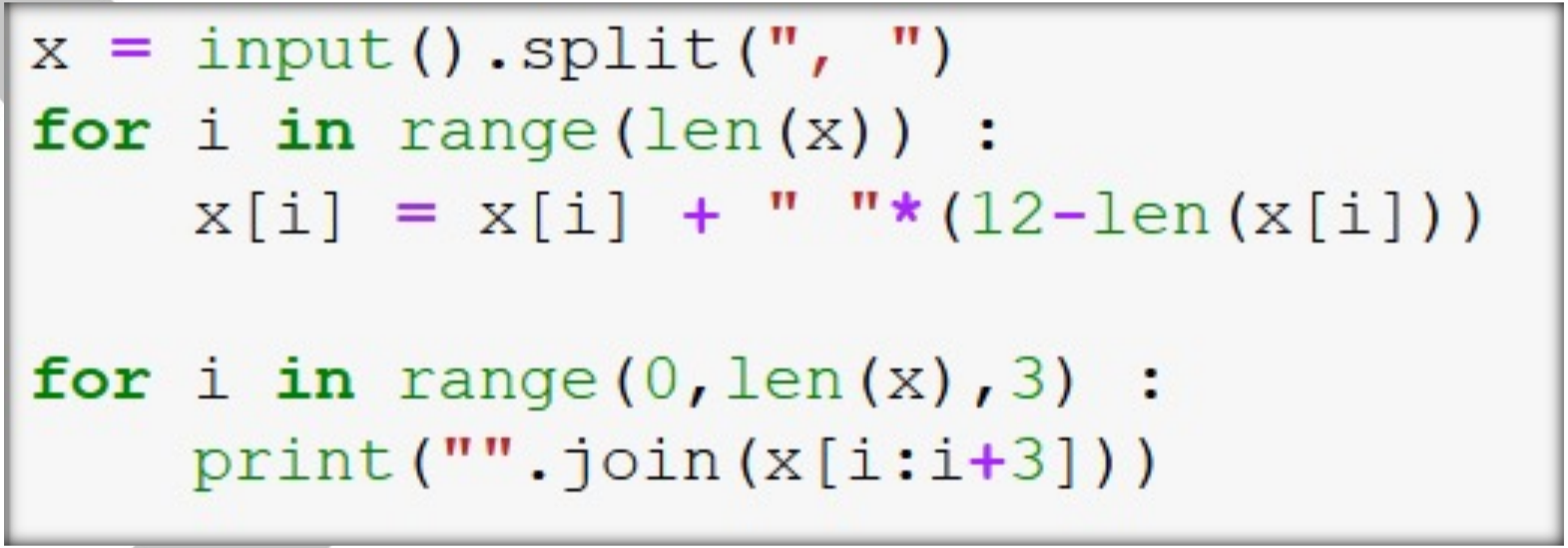
Each name must be displayed using 12 characters (add space when appropriate).

For example:
Bucciarati, Gioruno, Abbacchio, Arancia, Pannacotta, Mista, Resotto, Doppio will result in



```
Bucciarati  Gioruno     Abbacchio
Arancia     Pannacotta  Mista
Resotto     Doppio
```

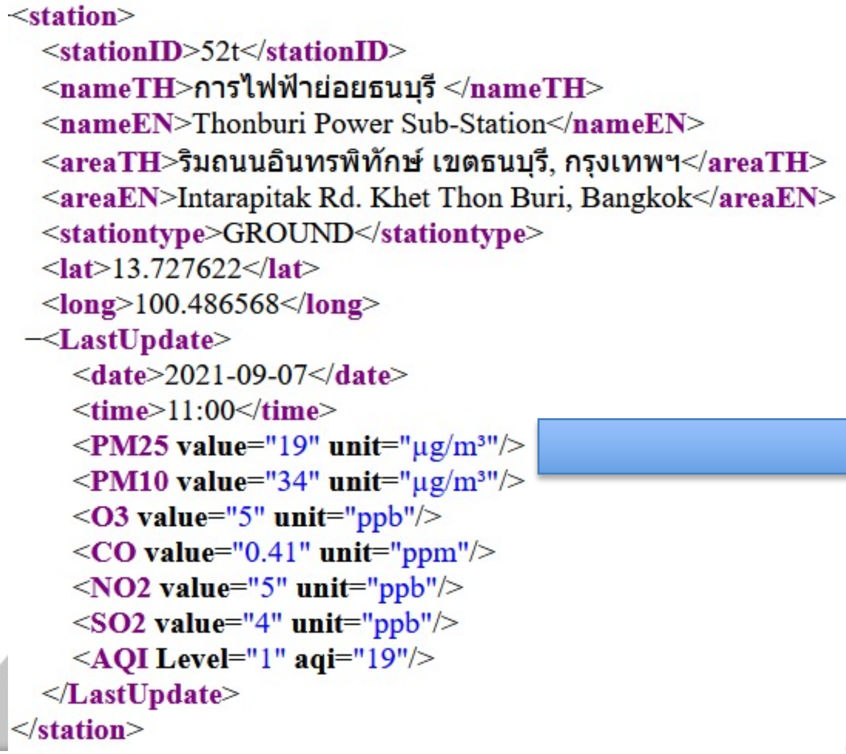
Example 1 (code)



```python
x = input().split(", ")
for i in range(len(x)) :
    x[i] = x[i] + " "*(12-len(x[i]))

for i in range(0,len(x),3) :
    print("".join(x[i:i+3]))
```

Example 2: Air Quality Report!!

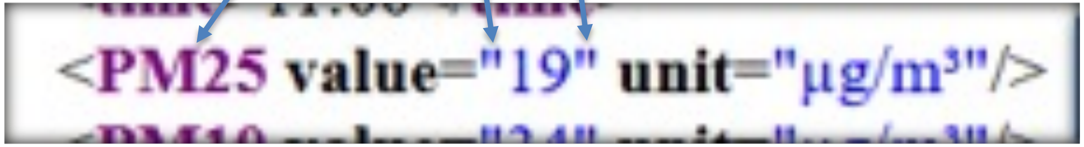
- http://air4thai.pcd.go.th/services/getNewAQI_XML.php?stationID=52t stores an XML file with air quality information of a site. The file is updated regularly.



```
<station>
  <stationID>52t</stationID>
  <nameTH>การไฟฟ้าย่อยธนบุรี </nameTH>
  <nameEN>Thonburi Power Sub-Station</nameEN>
  <areaTH>ริมถนนอินทรพิทักษ์ เขตธนบุรี, กรุงเทพฯ</areaTH>
  <areaEN>Intarapitak Rd. Khet Thon Buri, Bangkok</areaEN>
  <stationtype>GROUND</stationtype>
  <lat>13.727622</lat>
  <long>100.486568</long>
 -<LastUpdate>
    <date>2021-09-07</date>
    <time>11:00</time>
    <PM25 value="19" unit="µg/m³"/>
    <PM10 value="34" unit="µg/m³"/>
    <O3 value="5" unit="ppb"/>
    <CO value="0.41" unit="ppm"/>
    <NO2 value="5" unit="ppb"/>
    <SO2 value="4" unit="ppb"/>
    <AQI Level="1" aqi="19"/>
  </LastUpdate>
</station>
```

We want a program that reads this url and print the value of PM 2.5

Example 2 (Code)



```
<PM25 value="19" unit="µg/m³"/>
```

```
import urllib.request

def find(s, start, c):
    for i in range(start, len(s)):
        if s[i] == c: return i
    return -1

url =
"http://air4thai.pcd.go.th/services/getNewAQI_XML.php?station
ID=52t"
web = urllib.request.urlopen(url)
for line in web:
   line = line.decode()
   if "<PM25 value=" in line:
      i = find(line, 0, '"')
      j = find(line, i+1, '"')
      print("PM 2.5 =", line[i+1:j])
      break
```

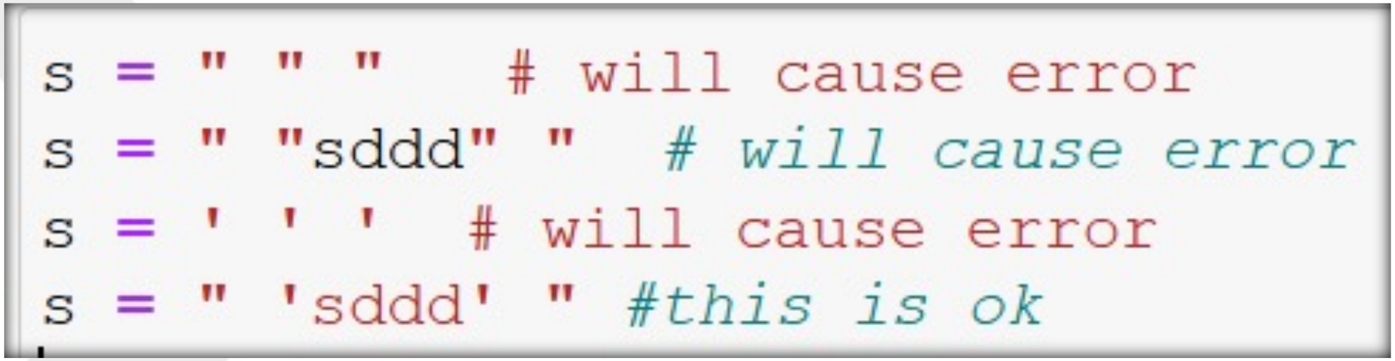
String already has this function!!

Exercise 7-1:

- See the Colab notebook.
- There are 5 questions.

Escape Characters

- S = “   “ cannot have “ inside
- S = ‘   ‘ cannot have ‘ inside



```
s = " " "    # will cause error
s = " "sddd" "   # will cause error
s = ' ' '   # will cause error
s = " 'sddd' " #this is ok
```

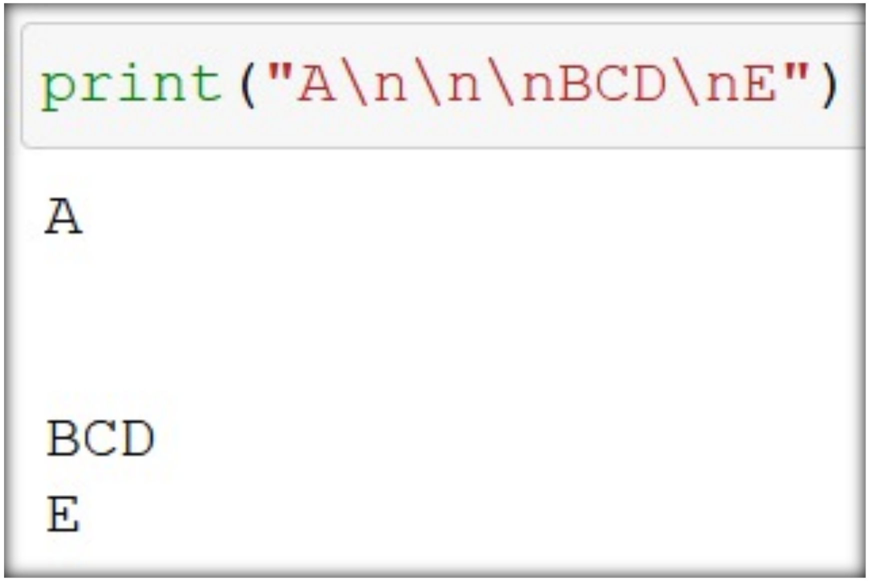
We solve it using back slash \

Using \

- `\"` means "
- `\'` means '
- `\\` means \
- `\n` means go to new line

Example: using \

```
s = "Hello"
print(s)          # Hello

s = "\"Hello\""
print(s)          # "Hello"

s = "\'Hello\'"
print(s)          # 'Hello'

s = "\"\'\\Hello\\"
print(s)          # "'\Hello\

s = "Hello\nPython"
print(s)                  # Hello
                          # Python
```



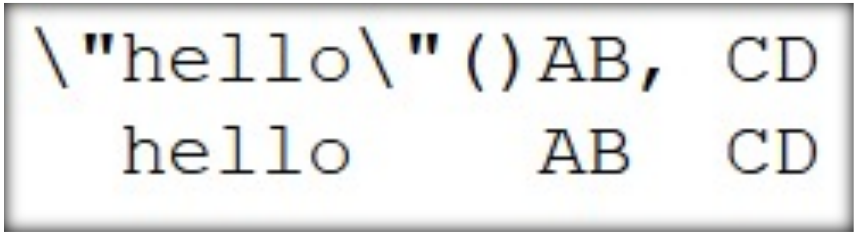
```
print("A\n\n\nBCD\nE")
```

```
A


BCD
E
```

Example: Replacing “ ‘ / \ ( ) , . : ; with space

Some don’t need a \

```
def remove_punc(t) :
    result = ""
    for e in t :
        if e in "\"\'/\\(),.:;" :
            result += " "
        else :
            result += e
    return result

s = input()
print(remove_punc(s))
```



```
\"hello\"()AB, CD
  hello      AB  CD
```

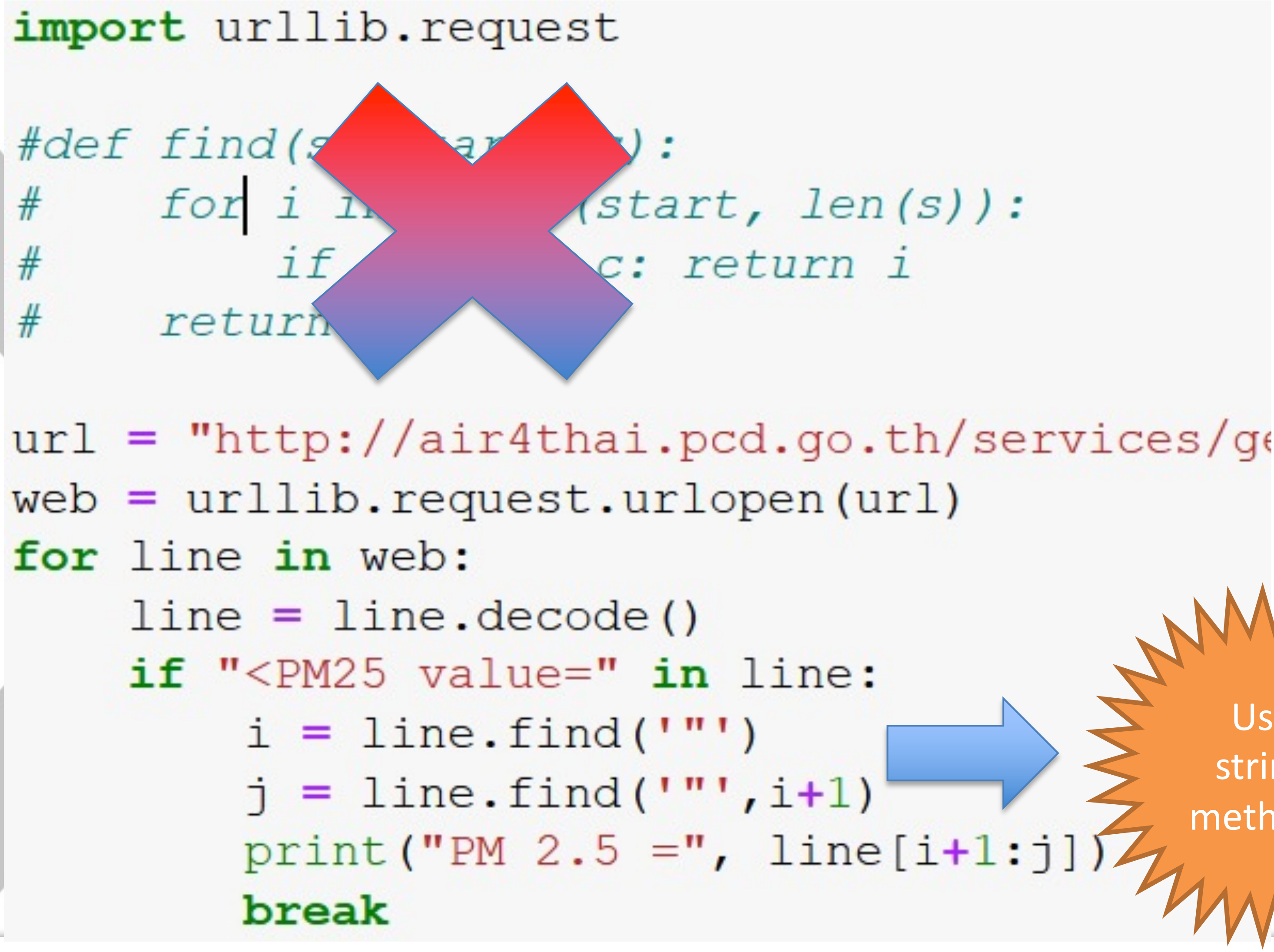
String Methods

Important: lower, upper, and strip return a new string!! Old string is unchanged

```
      01234567890123
s = "  Hello World "
```

- `len(s)` returns 14 , `len("")` returns 0
- `s.lower()` returns `"  hello world "`
- `s.upper()` returns `"  HELLO WORLD "`
- `s.strip()` returns `"Hello World"`
- `s.find("o")` returns 6 #first position
- `s.find("ex")` returns -1 # not found
- `s.find("o",7)` returns 9 #start at pos 7

Getting back to air quality example!



```
import urllib.request

#def find(s    ar     ):
#     for i i     (start, len(s)):
#         if          c: return i
#     return

url = "http://air4thai.pcd.go.th/services/ge
web = urllib.request.urlopen(url)
for line in web:
    line = line.decode()
    if "<PM25 value=" in line:
        i = line.find('"')
        j = line.find('"',i+1)
        print("PM 2.5 =", line[i+1:j])
        break
```

Use string method!

100th Anniversary of
Chula Engineering 2013

Exercise 7-2

- See Colab notebook

python
powered

Function Composition

```
x = math.radians(d)
s = math.sin(x)
y = abs(s)
r = round(y, 2)
```

```
r = round(abs(math.sin(math.radians(d))), 2)
```

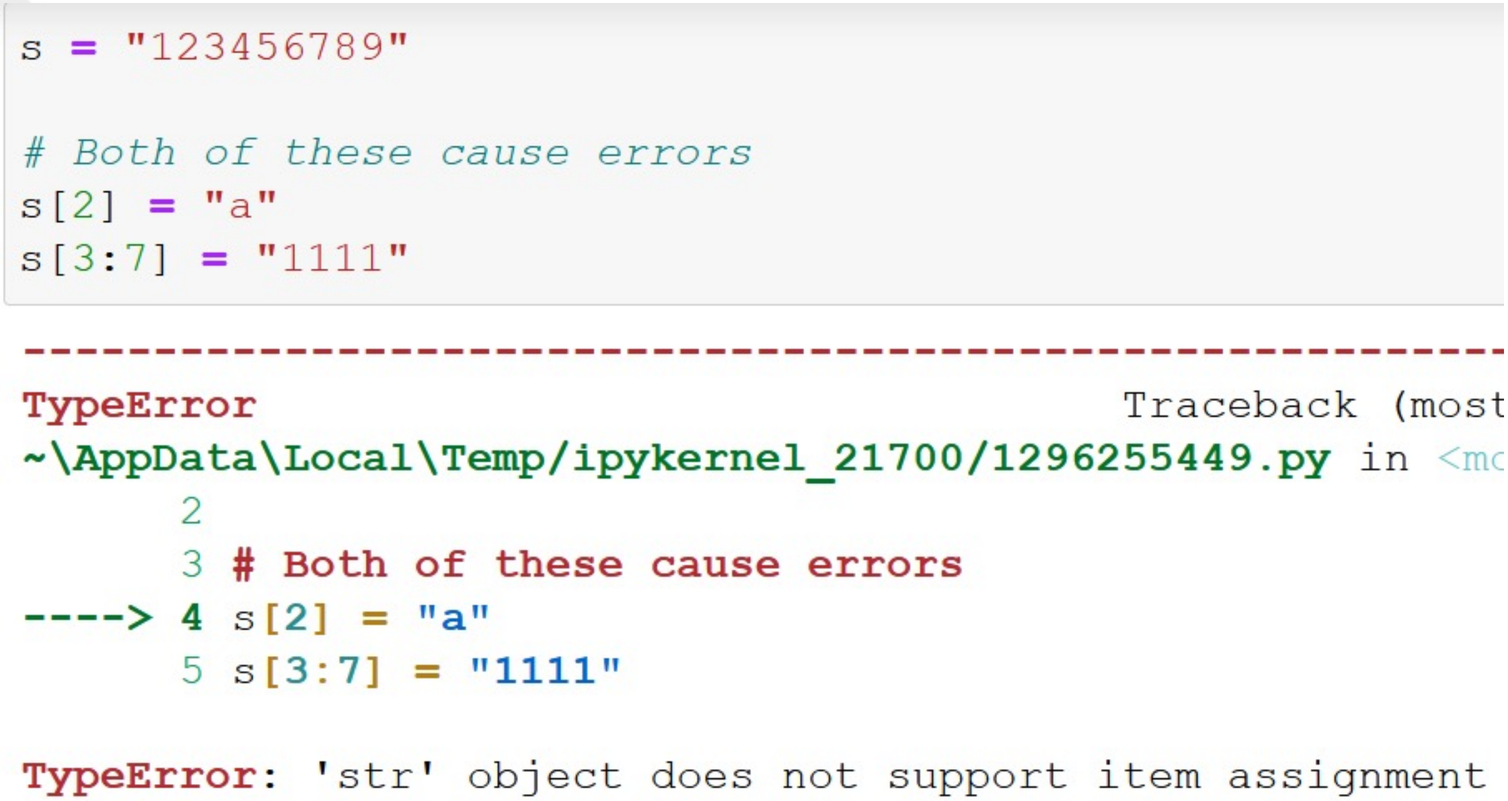
Method Chaining

```
line1 = input()
line2 = line1.strip()
line3 = line2.upper()
i     = line3.find("OK")
```

```
i = input().strip().upper().find("OK")
```

Beware

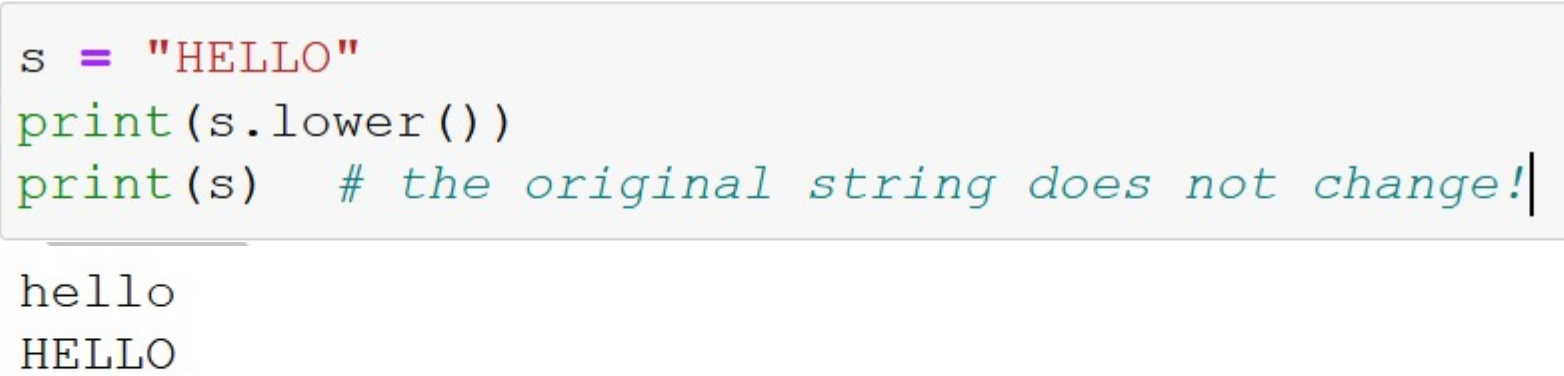
- String cannot be changed!



```
s = "123456789"

# Both of these cause errors
s[2] = "a"
s[3:7] = "1111"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most
~\AppData\Local\Temp/ipykernel_21700/1296255449.py in <mc
      2
      3 # Both of these cause errors
----> 4 s[2] = "a"
      5 s[3:7] = "1111"

TypeError: 'str' object does not support item assignment
```

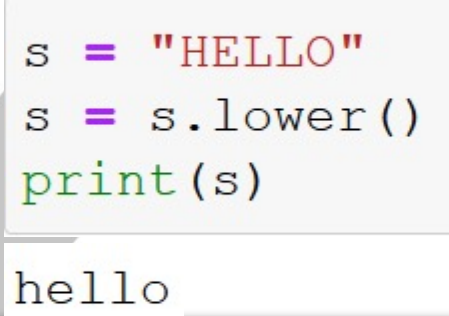
- String method therefore does not change the string. It creates a new string!



```
s = "HELLO"
print(s.lower())
print(s)   # the original string does not change!
```

```
hello
HELLO
```

- But we can re-assign the original string variable to store the method's result.

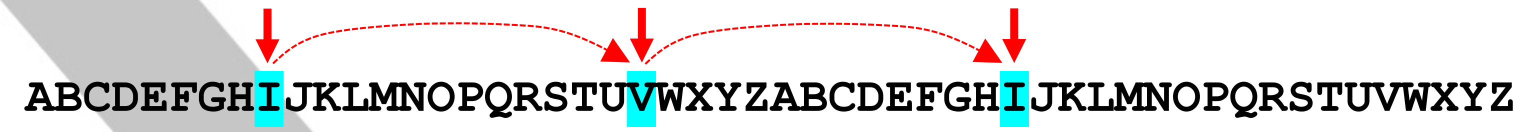


```
s = "HELLO"
s = s.lower()
print(s)
```

```
hello
```

Example: rot-13 (encode/decode)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z



I SEE TREES OF GREEN

RED ROSES TOO

I SEE EM BLOOM

FOR ME AND FOR YOU

AND I THINK TO MYSELF

WHAT A WONDERFUL WORLD

V FRR GERRF BS TERRA

ERQ EBFRF GBB

V FRR RZ OYBBZ

SBE ZR NAQ SBE LBH

NAQ V GUVAX GB ZLFRYS

JUNG N JBAQRESHY JBEYQ

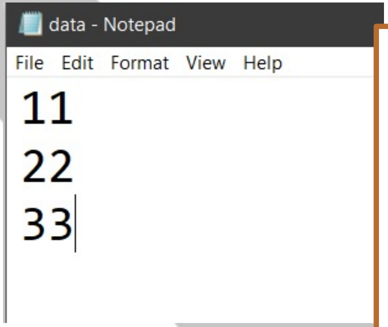
```
def rot_13(s):
    alphabets = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    alphabets *= 2
    rot13 = ""
    for ch in s :
        if "A" <= ch <= "Z" :
            pos = alphabets.find(ch)
            k = pos + 13
            rot13 += alphabets[k]
        else :
            rot13 += ch
    return rot13

original = "I HAVE A BAD FEELING ABOUT THIS."
print(rot_13(original))
```

```
V UNIR N ONQ SRRYVAT NOBHG
GUVF.
```

Exercise 7-3

- See Colab notebook

Reading data from a file

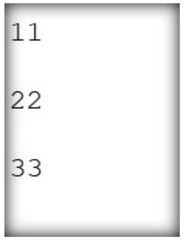


```
11
22
33
```

```
fn = open("data.txt", "r")  # r indicates READ mode
line1 = fn.readline()       # read one line (it can read beyond the last line)
line2 = fn.readline()
line3 = fn.readline()
line4 = fn.readline()

fn.close() # close file

print(line1)
print(line2)
print(line3)
print(line4)  # it doesn't have anything to print, but it does not cause error!
```

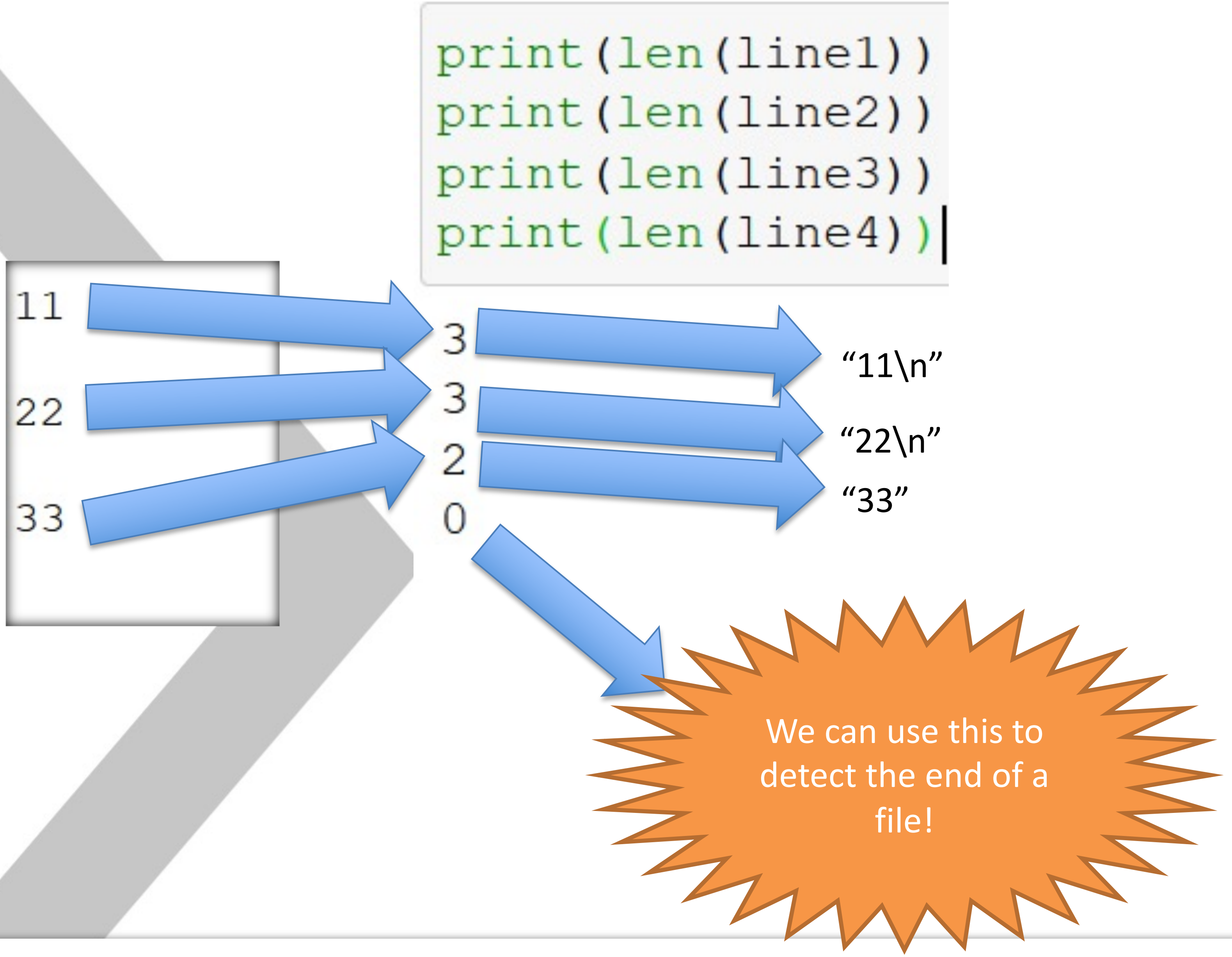
Otherwise,

- RAM reserved for working with the file will not be free!
- Other programs will be locked out from seeing the file forever.
- Also, writing to file does not really occur until you close it.



```
11

22

33
```

Seems to have too many “\n” ?



```
print(len(line1))
print(len(line2))
print(len(line3))
print(len(line4))
```

```
11
22
33
```

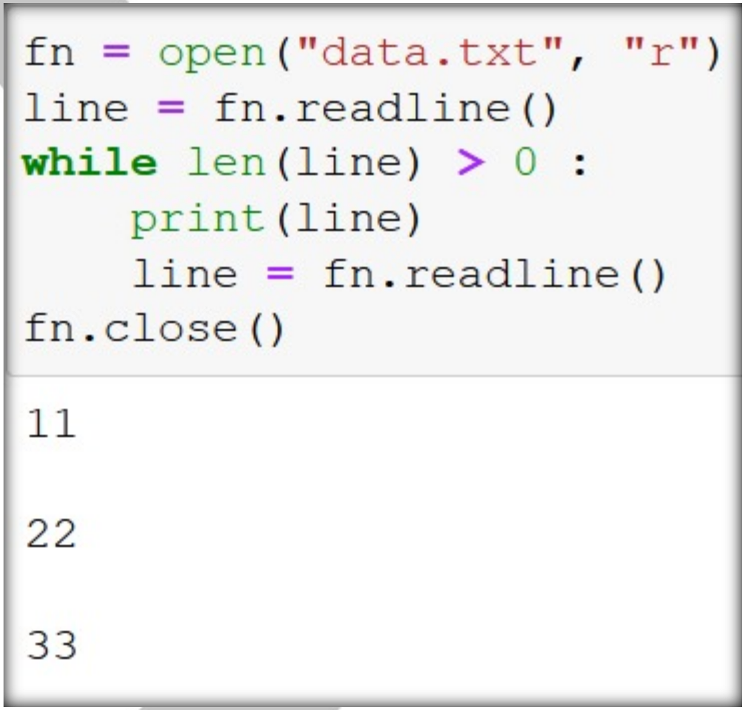
```
3
3
2
0
```

“11\n”

“22\n”

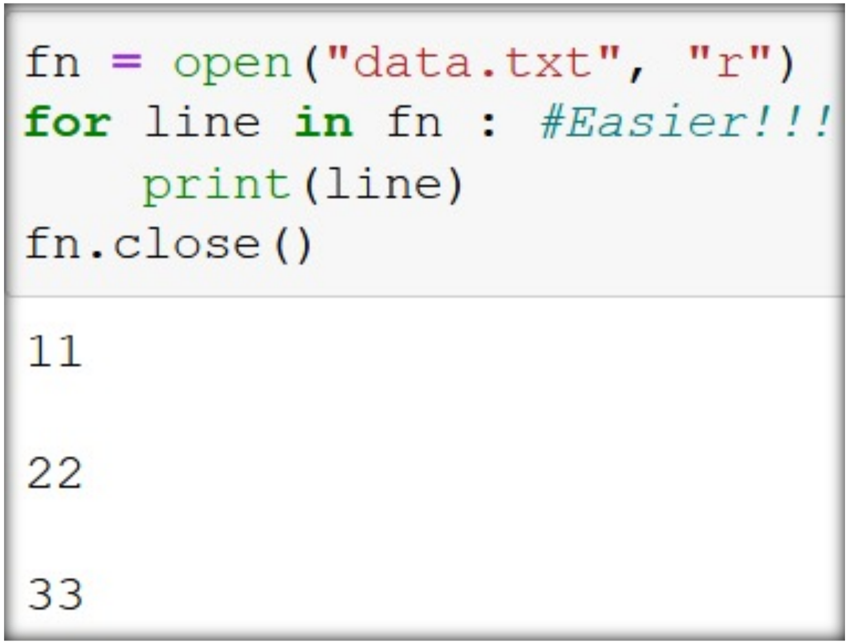
“33”

We can use this to detect the end of a file!



```python
fn = open("data.txt", "r")
line = fn.readline()
while len(line) > 0 :
    print(line)
    line = fn.readline()
fn.close()
```

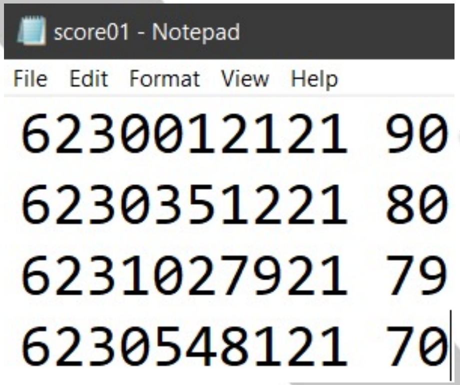
```
11

22

33
```



```python
fn = open("data.txt", "r")
for line in fn : #Easier!!!
    print(line)
fn.close()
```

```
11

22

33
```

Example: Print average score from the first 3 students in a file

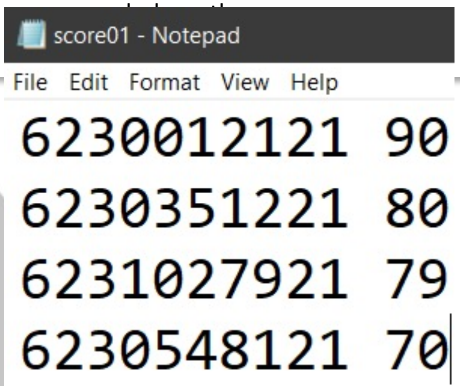


```
6230012121 90
6230351221 80
6231027921 79
6230548121 70
```

“\n”

```
fn = open("score01.txt", "r")
count = 0
sum = 0
i = 0
for line in fn :
    i += 1
    if i > 3 :
        break
    sum += float(line[-3:-1:1])
    count += 1
print("Average = ", sum/count)

fn.close()
```

This is not good if scores are different decimals!



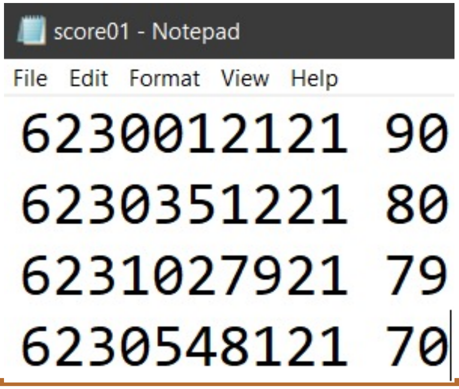
```
6230012121 90
6230351221 80
6231027921 79
6230548121 70
```

```
sum_points = 0; n = 3  # we can easily change n (very easy to see)

infile = open(“score01.txt", "r")
for k in range(n):
    line = infile.readline()
    x = line.split()
    sum_points += float(x[1])
infile.close()

print("Average =", sum_points/n)
```

[62......., 90]

Any decimal points are ok now

Example: sort scores from high to low



```
6230012121 90
6230351221 80
6231027921 79
6230548121 70
```

```
students = []

fn = open("score01.txt", "r")
for line in fn :
    sid,point = line.strip().split() # split gives 2 value in a list. We can use 2 variables.
    point = float(point)
    students.append([point, sid]) # add to a list, with score first, so it can be used in sort.
fn.close()

students.sort(reverse=True)   # we can also sort normally and then reverse the list.
for [point,sid] in students :
    print(sid,point)
```

Exercise 7-4

- Find average of students whose ID starts with a given number.

score02.txt

```
6230012121 90
6130351221 80
6231027921 79
5830548121 65
6031087221 70
6230550321 72
6230432721 87
6230215221 95
6130518321 72
```

Input

```
score02.txt 62
```

```
score02.txt 59
```

Output

```
Average = 84.6
```

```
No data
```

Writing to a file

```
fout = open(filename, "w")  #open for writing
fout.write("First line")    # write at the end of the file
fout.write("Text\n")        # write and go to a new line


fout.close()
```

Example: Record 100 numbers into a file, 10 numbers per line

- Triangular number is number + i , number starts from 0 and i starts from 1.
- 1 3 6 10 15 ... comes from 0+1, 1+2, 3+3, 6+4, 10+5, ...

```
fout = open("tri_numbers.txt", "w")  #open for writing
num = 0
for i in range(1,101) :
    num = num + i
    fout.write(str(num)+" ")
    if i %10 == 0 :
        fout.write("\n")
fout.close()
```

Exercise 7-5

- See Colab Notebook